C

ccline

A zsh plugin that turns your shell prompt into an AI assistant powered by Claude or Codex.

About

ccline is a zsh plugin that works like an AI command helper for your terminal: type a thought at the prompt and it sends the text to Claude or Codex for an answer. It hooks into zsh's command_not_found_handler, so normal typo handling stays fast for single-word mistakes while multi-word prompts can be routed to an LLM CLI. The result is rendered in Markdown, and any suggested shell commands can be picked and executed in your current shell session.

What is ccline?

ccline is an AI command helper for zsh that reuses `command_not_found_handler` to turn

switch to a browser or a separate chat app, it lets you ask from the shell itself. If you type a single word that is not a command, zsh still shows the normal error immediately. If you type two or more words, ccline treats it more like a question and sends it to your LLM CLI.

This makes ccline feel like a lightweight AI terminal assistant rather than a full shell replacement. The app is versioned as `v0.2.2`, installs with a one-line shell command, and is published as MIT licensed open source. It targets macOS users with zsh, and it expects either the `claude` CLI or the `codex` CLI to be available on your PATH.

Key features

ccline focuses on a small set of terminal-native behaviors that make an AI command helper practical instead of distracting.

- **Intercepts command-not-found events**: ccline plugs into `command_not_found_handler`, so it only steps in when zsh does not recognize the input.
- **Keeps typo handling fast**: one-word inputs like `gti` still fail normally, which avoids an unnecessary AI request for obvious mistakes.
- **Uses Claude or Codex automatically**: it prefers `claude` when available and falls back to `codex` if needed.
- **Supports Markdown rendering**: answers are shown with `glow` when installed, or with a built-in Perl renderer when it is not.
- **Lets you run suggested commands**: shell commands extracted from the response appear in an arrow-key picker, with Enter to execute and `q` to cancel.
- **Runs in your live shell**: selected commands execute via `eval`, so `cd`, `export`, aliases, functions, and shell history behave as if you typed them yourself.
- **Keeps setup minimal**: configuration is limited to environment variables such as `CCLINE_BACKEND` and `CCLINE_MODEL`.

The design goal is clear: make an AI command helper that stays close to the terminal workflow and does not add a big configuration surface.

How to use it

Make sure zsh is your shell, since ccline only hooks into zsh and does not support other shells.

Install one backend CLI: either `claude` or `codex`, authenticated and available on your PATH.

Optionally install `glow` if you want richer Markdown formatting in the terminal.

Add or edit your shell config so ccline is sourced from `~/.zshrc`.

Type a thought at the prompt, such as a multi-word request, and let the AI command helper return an answer.

Use the picker to choose a suggested command, run all commands, or cancel if you only wanted the explanation.

The app also supports two environment variables before the source line in `~/.zshrc`:

`CCLINE_BACKEND` to force `claude` or `codex`

`CCLINE_MODEL` to override the model name passed to the active backend

The install flow is designed to be repeatable, and rerunning the script should not duplicate the source line.

Who it is for

ccline is for people who already live in zsh and want an AI command helper inside the shell instead of in a separate tab. It is especially useful for:

developers who ask the terminal how to do something, then want to execute the suggested command immediately

macOS users who already rely on zsh as the default shell

Claude or Codex users who want a terminal-first workflow

power users who prefer short, direct interactions over a GUI chat client

people who want Markdown-formatted answers without leaving the shell

It is less useful for users who do not use zsh, do not have access to a supported LLM CLI, or want a browser-based chat experience. Because ccline runs commands in the live shell, it suits users who are comfortable reviewing output before pressing Enter.

ccline is intentionally narrow, and that comes with real tradeoffs.

- It only works in **zsh**; other shells are not supported.
- It depends on an external **Claude or Codex CLI**, so it is not a standalone hosted AI service.
- It executes selected commands with `eval` **in the current shell**, which is powerful but also means you should review suggested commands carefully.
- It is optimized for **interactive shell help**, not for batch automation, code generation pipelines, or general search.
- Markdown rendering improves readability, but the built-in Perl fallback is there only to avoid an extra dependency, not to turn the terminal into a full document viewer.
- For anything security-sensitive, production-impacting, or irreversible, an official command reference or vendor documentation is still the safer source than an AI command helper.

If you want an AI command helper that feels native to zsh and stays close to your prompt, ccline is a focused open-source option. It is best when you want a quick explanation, a likely command, and the ability to run it right away in your shell.

FAQ

What is ccline used for?

Is ccline free to use?

Which platforms does ccline support?

Does ccline work on shells other than zsh?

Does ccline require a paid account?

What do I need installed before I can use ccline?

How do you install ccline?

How does ccline decide when to send text to the AI?

Can I control which model ccline uses?

Will commands suggested by ccline affect my current shell session?

Sponsored

Partly Featured

Finance

Partly is a mobile app that lets friends quickly scan receipts, split any bill fairl...

Free # Finance

View ↗

Visit website ↗

ccline

Pricing	Free
Category	AI
Platform	Web, macOS, CLI
Launched	Aug 2026
Updated	today
Verified	2 months ago

ai # zsh # terminal # ai assistant # claude # codex # open source # shell

Dir Hub

A curated directory of small, well-made apps. Browse by category or search to find the right one in fewer clicks.

DISCOVER

Home

Pricing

Submit your app

COMPANY

About

Contact

LEGAL

Privacy

Terms

© 2026 Dir Hub

Curated indie apps